The Recipient Update Service Internal Filter Evaluator

By Marcelo Calbucci

Keywords: Recipient Update Service, Filter Evaluator, purportedSearch, msExchPurportedSearchUI, msExchEnableInternalEvaluator, LDAP Filter

The Recipient Update Service is a service made to apply policies to entries on the Active Directory, based that each entry match a specific Purported Search for that policy.

For example, you can create Recipient Policies that apply only to the board of directors of your company, or for only those whose country location is Brazil. The same way, you can create Address List that has only persons that have “manager” in their title or that the department is “Legal”.

Each of these rules is a Purported Search and it is stored on the “purportedSearch” attribute on the AD. If you create your own Recipient Policy or Address List policy that is not the one that is created by default, you’ll also see an attribute called msExchPurportedSearchUI, which is used to reconstruct the UI in order to let you change the filter later.

When the Recipient Update Service “sees” that an entry was added, what it does is to test this entry against each Policy and each Address List, and create a list of matches. Then it calls a provider to fill the attributes for that entry. All providers are embedded in the Recipient Update Service you cannot add new providers.

We have 5 providers internally:

1) Recipient Policy: Generate proxyAddresses and other mail attributes
2) Address List: Populate the showInAddressBook attribute
3) Hidden DL Membership: Add Non-Canonical ACEs to the Security Descriptor for Groups that have Hide DL Membership checked on the UI.
4) Mail Enable Recipient: Add legacyExchangeDN and displayName for mail-enabled objects
5) Mailbox Enabled User: Add legacyExchangeDN, msExchHomeServerName, homeMDB, homeMTA, msExchMailboxGuid and displayName for mailbox enabled users.

The last 3 providers are called the “System Policies” and you cannot edit them thru the User Interface, nor you should. You can know more about it on the KB article “The System Policies for the Recipient Update Service”

For the two first providers, you can (and probably will) add, modify and delete new policies. By default, the following policies/address lists are created when you install Exchange 2000:

- Recipient Policy:
  - Default Policy
- System Policies:
  - Hidden DL Membership
  - Mail Enable Recipient
  - Mailbox Enable User
- Address Lists:
  - Default Global Address List (a.k.a. GAL)
  - All Conferencing Resources
  - All Contacts
  - All Groups
  - All Users
  - Public Folders

Each one of this has its own purportedSearch. For example, the “Default Policy” has the purportedSearch equal to “(mailNickname=*)”, which is pretty simple, and the “Mailbox Enable User” System Policy has a much more complex purportedSearch:

```
“(|(&(objectCategory=person)(objectClass=user)(mailnickname=*)
(msExchHomeServerName=*))(&(objectCategory=person)
(objectClass=user)(mailnickname=*)(homeMdb=*))
(&(objectCategory=person)(objectClass=user)(mailnickname=*)
(homeMta=*)))”
```

Total, you’ll have 10 purportedSearches to be tested for each new user that you create on the Active Directory. It could be 11 if you also have an Exchange 5.5 on the same site, or more if you create your own Recipient Policies and Address Lists.

To evaluate if an entry “meets” a purportedSearch we can call the Active Directory, doing a base-search on the DN of the entry that we want to test, and passing as filter the value of the purportedSearch. If the result is the entry itself, then it means that we meet the purportedSearch, otherwise, if the result is empty, we don’t.

This is a very effective and safe way to perform this test, but it means that for each new entry we have to do 10 searches on the Active Directory. If you have a total of 20,000 entries in your directory, it would mean at least 200,000 searches. And this is a lot!

The solution to that is to implement a subsystem that simulates the Active Directory LDAP Filter Evaluator. However, this is a very hard task, and it will not be guaranteed that the behavior will be

identical, since Windows 2000 can change the behavior of their Filter Evaluator, and we would have to change ours too.

But a good part of the AD Filter Evaluator is based on a Standard, thus, guaranteed to work as expected.

So, what we did is to implement a limited LDAP Filter Evaluator that would improve the Recipient Update Service by several folds. This filter is only enabled by default for the 10 purportedSearches above. It is disabled to any custom Recipient Policy or Address List.

This is determined by reading the attribute "msExchPurportedSearchUI" and "msExchEnableInternalEvaluator" for each Policy or Address List. The algorithms is like that:

```
If msExchEnableInternalEvaluator is present
        If msExchEnableInternalEvaluator = = TRUE Then
                Use Internal Filter Evaluator
        Else
                Don't use Internal Filter Evaluator
        End If
Else
        If msExchPurportedSearchUI is present Then
                Don't use Internal Filter Evaluator
        Else
                Use Internal Filter Evaluator
        End If
End If
```

With this algorithm you can always forcibly enable or disable the Internal Filter Evaluator for any Policy or Address List, being it custom or not

As we said, the Internal Filter Evaluator doesn't behave 100% identical to the Active Directory LDAP Filter Evaluator, but it is very close. Instead of writing what the Internal Filter Evaluator is compatible with, we'll tell you what is not compatible in the LDAP filter:

- Case-sensitive comparisons;
- Match rule;
- Ambiguous Name Resolution (ANR) and

First, the Internal Filter Evaluator only supports case-insensitive comparisons. This means that if you have a filter rule like

(displayNamePrintable=JOHN*),

it will evaluate to true entries that have "JOHN*" and "john*" on this attribute.

The Match Rule is something that let you do more operations in an LDAP search than simple those like and, or, not, equal, greater-than and less-than. It let you specify a filter that says that you want

the third character of a string to be ‘c’. All match rules are not supported on the Internal Filter Evaluator.

Finally, Ambiguous Name Resolution, more known as ANR is a way to specify a search for common attributes. For example, if you specify “(ANR=John)” in a filter, and search the AD, it will return entries that have the first name that starts with “John”, the mailNickname equal “John”, the last name that starts with “John” and a few other attributes. This can be configured by the Schema and a set of heuristics attributes. Again, the Internal Filter Evaluator does not support ANR.

You should use be switching for the Internal Filter Evaluator if you have a very large Active Directory with several custom Recipient Policies and Address List, otherwise there is not much gain.